

# PERANCANGAN SISTEM OPTIMASI PENJADWALAN PENGANGKUTAN SAMPAH DI SURABAYA SECARA ADAPTIF MENGGUNAKAN METODE ALGORITMA GENETIKA

Febrianto Nur Anwari, Arna Fariza, S.Kom, M.Kom, Entin Martiana S.Kom, M.Kom,

Ira Prasetyaningrum S.Si,MT.

*Jurusan Teknik Informatika, PENS - ITS Surabaya*

*Jl. Raya ITS, Surabaya*

*+62(31) 594 7280; Fax: +62(31) 594 6114*

**E-mail :** [febrizio@student.eepis-its.edu](mailto:febrizio@student.eepis-its.edu), [arna@eepis-its.edu](mailto:arna@eepis-its.edu), [entin@eepis-its.edu](mailto:entin@eepis-its.edu), [ira@eepis-its.edu](mailto:ira@eepis-its.edu)

## ABSTRAK

*Sistem Pengangkutan sampah di Kota Surabaya dibagi menjadi dua bagian, yaitu sistem pengangkutan dari Kelurahan / Kecamatan (KK) ke LPS kemudian dari LPS ke LPA. Untuk melaksanakan sistem pengangkutan ini di setiap harinya telah disediakan gerobak tarik untuk mengangkut sampah dari KK ke LPS. Lalu ada juga armada truk pengangkut sampah yang disediakan oleh Dinas Kebersihan dan Pertamanan untuk mengangkut sampah dari LPS ke LPA. Setiap harinya, armada-armada yang bekerja dari LPS ke LPA ini selalu bekerja secara teratur dan sesuai dengan perintah yang diberikan. Namun, rute-rute yang dilewati dan jadwal pengangkutan tiap hari bisa berubah-ubah dan sesuai dengan keinginan sopir dari armada yang bersangkutan. Oleh karena itu, terkadang sampah masih menumpuk di satu LPS karena sampah yang datang dari KK lebih banyak daripada biasanya, sedangkan kinerja armada pengangkutan tidak bisa menyesuaikan dengan kondisi tersebut.*

*Oleh karena itu, saya mengusulkan pada paper ini sebuah sistem optimasi penjadwalan tersebut dengan menggunakan Algoritma Genetika. Masing-masing solusi direpresentasikan kedalam tiap-tiap individu genetik dalam algoritma tersebut yang mana tiap individu merepresentasikan sebuah solusi rute pengangkutan. Kemudian dengan melakukan beberapa operasi seperti seleksi, cross over dan mutasi pada individu-individu tersebut, bisa menghasilkan individu baru dengan nilai fitness yang semakin tinggi. Nilai fitness yang semakin tinggi artinya solusi tersebut adalah solusi paling optimal yang bisa didapatkan, yang merepresentasikan rute terpendek yang bisa dilalui oleh truk armada pengangkutan sampah. Selain itu, untuk mempermudah penyampaian informasi digunakanlah Sistem Informasi Geografis (SIG) dengan menggunakan teknologi Google Maps API yang bisa menunjukkan rute-rute yang bisa dilewati untuk mengambil sampah di titik-titik LPS tersebut hingga dibawa ke LPA. Hasil dari sistem optimasi ini menunjukkan bahwa satu truk armada bisa digunakan untuk mengangkut sampah lebih dari satu LPS, asalkan volume sampah tidak melebihi kapasitas truk. Dengan demikian, sistem ini bisa menekan biaya bahan bakar untuk proses pengangkutan sampah sehari-hari.*

**Kata Kunci :** *Sistem Optimasi, SIG, Algoritma Genetika, Rute, Sampah*

## 1. PENDAHULUAN

Sampah adalah salah satu masalah yang cukup pelik bagi beberapa kota besar seperti di Surabaya ini. Volume sampah yang cukup besar dan tidak sesuai dengan kapasitas Lahan Pembuangan di wilayah Kecamatan dan Kelurahan (KK) serta di Lahan Pembuangan Sementara (LPS) dan Lahan Pembuangan Akhir (LPA) adalah salah satu masalahnya. Bukan sebuah hal yang mengherankan, karena memang sebuah kota yang besar tentu memiliki jumlah penduduk yang besar pula. Katakanlah jika sebuah rumah yang berisi 4 orang menghasilkan sampah sebanyak  $0.5 \text{ m}^3$  perhari, maka bisa dibayangkan berapa volume sampah yang dihasilkan kota ini setiap harinya.

Permasalahan sampah tidak hanya pada volumenya saja. Kurangnya kinerja antar instansi-instansi terkait dalam mengelola sampah ini juga masih menjadi masalah. Meskipun kinerjanya bisa dikatakan cukup baik, tetapi masih saja terjadi penumpukan sampah di suatu KK dan LPS sehingga mencemari udara disekitarnya. Untuk itu, diperlukan sebuah sistem yang baik agar pengangkutan sampah dari KK ke LPS kemudian dilanjutkan ke LPA lebih teratur, sehingga bisa mengurangi resiko pencemaran udara yang disebabkan penumpukan sampah tersebut.

Untuk mengatasi hal tersebut, kami mencoba untuk merancang sebuah aplikasi perangkat lunak (software) yang bisa digunakan untuk mengoptimasikan penjadwalan pengangkutan sampah tersebut. Sistem yang kami buat ini terdiri dari dua

bagian pokok. Yang pertama adalah sistem optimasi dari LPS ke LPA, dan yang kedua adalah simulasi optimasi tersebut dengan memanfaatkan Google Maps API. Dengan menggunakan software ini, diharapkan sistem penjadwalan dalam manajemen dan pengelolaan sampah ini bisa lebih maksimal.

Sarwoko Mangkoedihardjo [2] dalam makalahnya yang berjudul “Peningkatan Kualitas Lingkungan Perkotaan: Pengelolaan Sampah dalam Perspektif Berkelanjutan” menjelaskan bagaimana proses pengolahan sampah di Surabaya dan apa saja infrastrukturnya. Dia menyebutkan bahwa Sistem umum manajemen sampah Surabaya saat ini melingkup area kota dengan pembuangan akhir sentralistik. Simpul-simpul sistemnya terdiri dari 5 jenis fasilitas pembuangan, yang kesemuanya terbuka untuk pembuangan sampah campuran. Pertama adalah bak sampah (BS) yang dikelola oleh masyarakat. Fasilitas kedua, yaitu pengangkutan sampah dari BS ke lahan pembuangan sementara (LPS). Bentuk fasilitasnya setara dengan gerobak tarik (GT). Jumlah GT per RW sekitar 1 – 2 unit, jumlah RW per kelurahan antara 5 – 7 dan dengan jumlah 168 kelurahan maka diperkirakan terdapat GT sejumlah 1.500 unit.

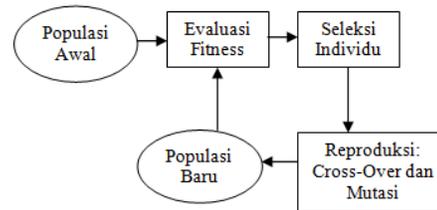
Ketiga adalah LPS, yang dikelola Dinas Kebersihan. LPS dapat berupa depo berukuran luas rata-rata 250 m<sup>2</sup> dengan luas lahan aktif operasional pembuangan sekitar 150 m<sup>2</sup> dan berupa landasan berukuran luas rata-rata 100 m<sup>2</sup>, yang dapat dioperasikan penuh. Keempat adalah fasilitas pengangkutan sampah dari LPS ke lahan pembuangan akhir sampah (LPA). Truk angkutan sampah (TA) adalah bentuk umum fasilitasnya. Ini berarti tiap TA melayani sekitar 2 LPS, atau 120 m<sup>3</sup>/hari harus diangkut. Jika tiap truk bermuatan penuh 12 m<sup>3</sup> dan beroperasi 3 kali/hari, maka pengosongan sampah dari LPS ke LPA hanya sebesar 36 m<sup>3</sup>/hari, atau 30 % dari seharusnya. Kelima adalah fasilitas LPA itu sendiri, yang dikelola Dinas Kebersihan. Cara pembuangan yang dilakukan adalah hampan terbuka (open dumping).

Berdasarkan penelitian diatas dan juga penjelasan mengenai macam-macam tempat pembuangan sampah tersebut, kami mencoba untuk membuat sebuah sistem yang bisa melakukan optimasi dalam pengangkutan sampah dari LPS ke LPA ini.

## 2. ALGORITMA GENETIKA

Menurut Wikipedia Indonesia [1] Algoritma Genetika adalah teknik pencarian yang di dalam ilmu komputer untuk menemukan penyelesaian perkiraan untuk optimisasi dan masalah pencarian. Algoritma

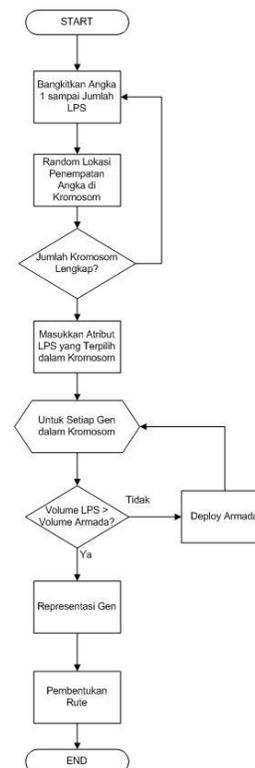
genetik adalah kelas khusus dari algoritma evolusioner dengan menggunakan teknik yang terinspirasi oleh biologi evolusioner seperti warisan, mutasi, seleksi alam dan rekombinasi (atau *crossover*). Sehingga untuk menyelesaikan masalah yang berhubungan dengan optimasi suatu pekerjaan, menggunakan algoritma genetika adalah salah satu cara yang terbaik.



Gambar 1: Alur umum Algoritma Genetika

### a. Alur Representasi Genetika

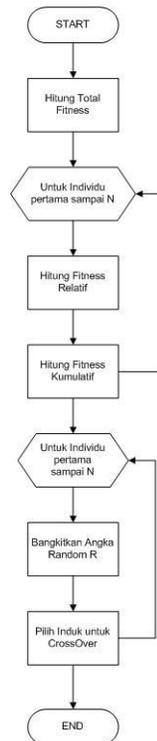
Representasi genetika sangat penting dalam algoritma genetika karena sangat berpengaruh terhadap solusi yang bisa didapatkan nantinya. Secara garis besar, representasi genetika dalam sistem ini bisa digambarkan sebagai berikut:



Gambar 2: Alur Representasi Genetika

### b. Alur Strategi Seleksi dengan Roulette Wheel

Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana offspring terbentuk dari individu-individu yang terpilih tersebut [3]. Dalam sistem ini, strategi seleksi menggunakan Roulette Wheel yang dijelaskan dalam Gambar 3



**Gambar 3:** Strategi Seleksi dengan Roulette Wheel

Algoritma untuk melakukan seleksi dengan Roulette Wheel diatas dapat dijelaskan sebagai berikut:

1. Hitung total fitness (F):  

$$\text{TotFitness} = \sum F_k; k = 1, 2, \dots, \text{popsize}$$
2. Hitung fitness relatif setiap individu:  

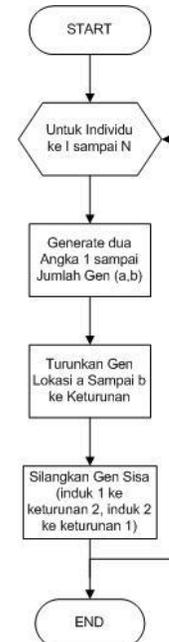
$$P_k = F_k / \text{TotFitness}$$
3. Hitung fitness kumulatif:  

$$q_1 = p_1$$

$$q_k = q_{k-1} + p_k; k = 2, 3, \dots, \text{popsize}$$
4. Pilih induk yang akan menjadi kandidat untuk di-crossover dengan cara:
  - Bangkitkan bilangan random r
  - Jika  $q_k \leq r$  dan  $q_{k+1} > r$  maka pilih kromosom ke k+1 sebagai kandidat induk

### c. Alur CrossOver dengan Metode OX

Untuk melakukan rekombinasi, dalam sistem ini digunakan proses rekombinasi Order CrossOver atau yang lebih dikenal dengan Metode OX. Untuk alurnya bisa dilihat dalam Gambar 4:



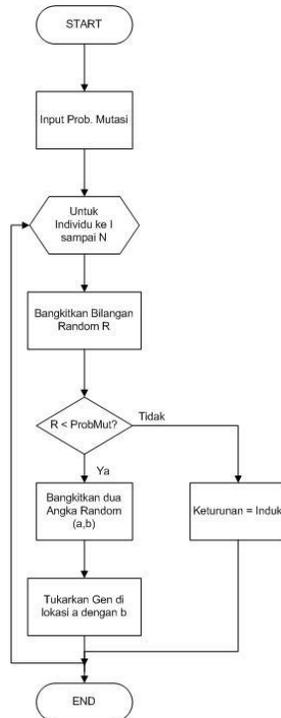
**Gambar 4:** Alur Reproduksi OX

Algoritma dari OX sendiri adalah:

1. Tentukan probabilitas crossover yang diinginkan (biasanya antara range 0.75 sampai 0.9).
2. Untuk setiap populasi sebanyak jumlah populasi lakukan:
  - a. Generate sebuah bilangan random antara nol sampai satu.
  - b. Jika bilangan tersebut lebih kecil daripada probabilitas crossover, maka lakukan crossover.
  - c. Sedangkan apabila bilangan tersebut lebih besar, maka induk 1 dan induk 2 langsung turun menjadi anak 1 dan anak 2.
3. Untuk setiap pasangan yang melakukan crossover lakukan:
  - a. Generate dua buah bilangan antara satu sampai jumlah gen setiap individu.
  - b. Turunkan gen induk di lokasi antara dua bilangan tersebut ke gen anak.
  - c. Silangkan gen induk yang belum diturunkan ke anak ke anak yang berlawanan. Jadi, turunkan gen induk pertama ke anak kedua, dan turunkan gen induk kedua ke anak pertama.

#### d. Strategi Mutasi dengan Teknik Swapping

Secara umum, strategi mutasi dengan menggunakan teknik swapping ini bisa dijelaskan dalam flowchart pada Gambar 5:



**Gambar 5:** Alur Strategi Mutasi dengan Teknik Swapping

Sedangkan untuk algoritmanya bisa dijelaskan sebagai langkah-langkah berikut:

1. Bangkitkan dua bilangan random (antara 0 sampai panjang kromosom) untuk setiap individu, sebanyak jumlah individu dalam populasi.
2. Tukarkan gen dengan lokasi kedua bilangan tersebut dengan cara:  
 $\text{bilanganTemp} = \text{gen1};$   
 $\text{gen1} = \text{gen2};$   
 $\text{gen2} = \text{bilanganTemp};$

### 3. TEKNOLOGI GOOGLE MAPS API

Google Maps adalah layanan gratis Google yang cukup populer. Kita dapat menambahkan fitur Google Maps dalam web Kita sendiri dengan Google Maps API. Google Maps API adalah library JavaScript. Yang dibutuhkan dalam penggunaan Google Maps API adalah pengetahuan tentang HTML dan JavaScript, serta koneksi Internet. Dengan menggunakan Google Maps API Kita dapat menghemat waktu dan biaya Kita untuk membangun aplikasi peta digital yang handal, sehingga Kita dapat focus hanya pada data-data. Biarkan data peta dunia menjadi urusan Google saja.

Saat ini versi terakhir Google Map API adalah versi 3. Versi ini, katanya, akan tampil lebih cepat dari versi sebelumnya khususnya untuk browser ponsel.

Untuk mulai menggunakan Google Maps API, kita bisa melakukan beberapa langkah berikut ini:

1. Memasukkan Maps API JavaScript ke dalam HTML kita.
2. Membuat element div dengan nama `map_canvas` untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan property- properti pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event `onload`.

Contoh penggunaan Google Maps API bisa kita lihat dalam gambar berikut ini. Disini, Google Maps API digunakan untuk menampilkan marker secara acak disebuah peta.[4]

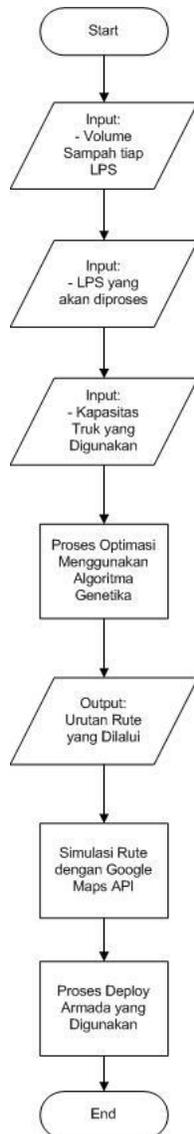


**Gambar 6:** Contoh penggunaan Google Maps API

### 4. PERANCANGAN SISTEM

Cara kerja sistem ini akan dibagi ke dalam beberapa tahapan proses supaya dapat dilihat dengan jelas. Tahapan proses dibagi menjadi beberapa tahapan sebelum tercipta sebuah sistem yang bisa digunakan untuk mendapatkan optimasi pengangkutan sampah yang paling optimal.

Sistem ini terbagi menjadi dua bagian poko, yaitu pembuatan sistem optimasi yang menggunakan Algoritma Genetika, dengan simulasi rute yang menggunakan teknologi peta dari Google Maps API. Flowchart umum perancangan sistem optimasi yang kami buat bisa dilihat pada Gambar 7 berikut.



**Gambar 7:** Alur umum sistem optimasi

Keterangan:

1. Sistem ini mengatur jalur pengangkutan sampah dari LPS ke LPA.
2. Untuk jalur pengangkutan, maka sistem akan memilih jalur yang baik untuk dilewati armada pengangkut sampah.
3. Rute yang ditempuh setiap hari selalu berubah-ubah secara adaptif, tergantung kepada volume sampah di satu LPS.
4. Input dari sistem optimasi ini adalah lokasi LPS, dan volume sampah yang ada di LPS tersebut. Setelah menerima input tersebut, sistem akan memberikan saran berapa jumlah armada yang harus diterjunkan untuk mengangkut sampah tersebut Hasil ini akan disinkronisasikan dengan sistem yang menangani jalur pengangkutannya, dan kemudian akan memberikan output berapa

kali sampah tersebut harus diangkut, beserta jalur yang harus dilalui armada dari LPS menuju ke LPA.

5. Sistem ini dituntut untuk adaptif mengikuti volume sampah yang tentu berubah-ubah setiap harinya. Sedangkan untuk jadwal pengangkutan, ditentukan secara manual oleh admin sistem ini.
6. Sistem juga bisa digunakan untuk perhitungan dan proses pembagian tugas armada berdasarkan rute yang harus dilalui pada hari itu.

## 5. PERANCANGAN SIG DENGAN GOOGLE MAPS API

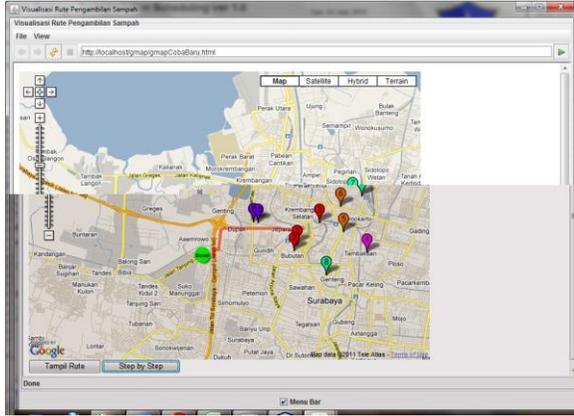
Untuk mensimulasikan rute hasil optimasi yang telah dikeluarkan oleh sistem optimasi, maka digunakan salah satu fitur dari Google Maps API yang disebut dengan Get Direction. Karena Google Maps API menggunakan arsitektur Javascript, sedangkan program optimasi menggunakan Java Desktop, maka alur untuk mensimulasikan rute adalah seperti yang terlihat pada Gambar 8:



**Gambar 8:** Alur Simulasi Rute dengan Google Maps

Alur diatas dibuat karena Google Maps API tidak bisa berkomunikasi secara langsung dengan MySQL, sehingga kita harus mengkonversikannya terlebih dahulu kedalam bentuk XML. Dari XML

yang kita buat itulah, kita bisa mengkomunikasikan Google Maps API dengan data hasil optimasi, sehingga Google Maps API bisa mensimulasikan rute sesuai dengan request dari sistem optimasi. Hasil dari Simulasi Sistem Optimasi contohnya seperti pada Gambar 8:

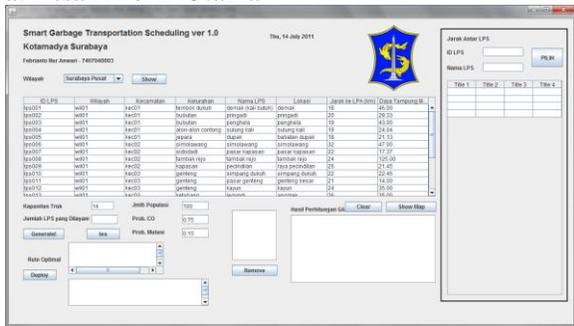


Gambar 8: Simulasi Rute dengan Google Maps API

## 6. PERANCANGAN USER INTERFACE

Di bagian ini akan membahas bagaimana perancangan user interface sistem optimasi, meliputi juga bagaimana interface dari laporan yang dihasilkan.

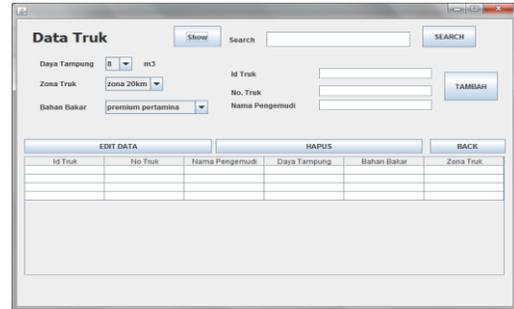
### a. Desain Form Utama



Gambar 9: Desain Form Utama

Form ini digunakan untuk mengakomodasi semua transaksi yang ada. Di Form ini, user bisa meng-inputkan parameter-parameter yang digunakan, seperti jumlah LPS yang akan dilayani, kapasitas truk yang digunakan, serta parameter-parameter yang dibutuhkan dalam Algoritma Genetika, seperti jumlah populasi, jumlah generasi, probabilitas Cross Over dan Probabilitas Mutasi.

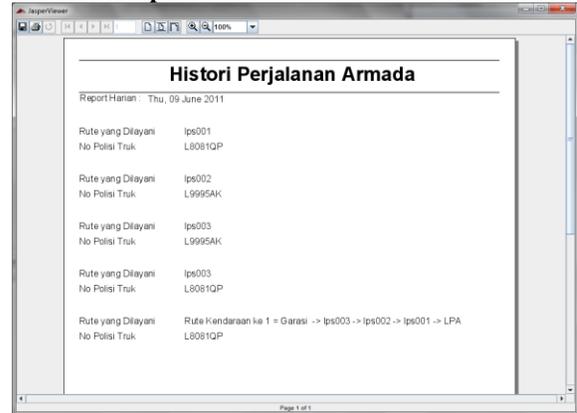
### b. Desain Form Master



Gambar 10: Desain Form Master

Form master ini digunakan untuk meng-inputkan berbagai macam data master, seperti master truk, yang mana digunakan untuk meng-inputkan data-data truk yang digunakan seperti nama pengemudi, dan nomor polisinya.

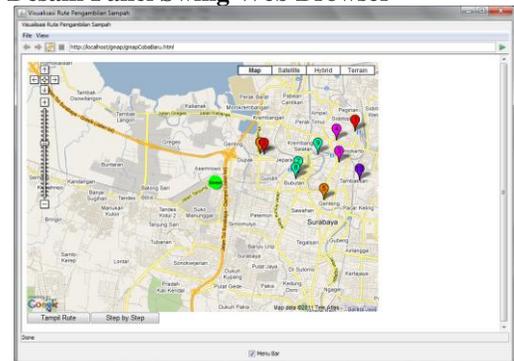
### c. Desain Report



Gambar 11: Desain Report

Untuk membuat report, digunakan plugin iReport milik Netbeans, dan juga library jasper. Report digunakan untuk menampilkan informasi histori, dan juga bisa dicetak untuk dijadikan hardcopy rekap harian.

### d. Desain Panel Swing Web Browser



Gambar 12: Desain Panel Swing Web Browser

Panel ini digunakan untuk menampilkan simulasi rute yang dihasilkan oleh sistem oprimasi. Panel ini menjembatani antara aplikasi simulasi rute yang dihasilkan oleh Google Maps API via Javascript dengan sistem optimasi yang berbasis Java Desktop. Panel ini menggunakan engine web browser milik Internet Explorer.

### 7. HASIL UJI COBA

Uji coba dilakukan dengan menggunakan sampel data 10 titik LPS sebagai berikut, dan dilakukan dalam sepuluh kali percobaan untuk diambil hasil terbaiknya:

**Tabel 1:** Data Uji Coba 10 LPS

	id_lps	nama_lps	jarak_lps	daya_tampung
<input type="checkbox"/>	lps001	demak (kali butuh)	18	46.00
<input type="checkbox"/>	lps002	pringadi	20	29.33
<input type="checkbox"/>	lps003	penghela	19	43.00
<input type="checkbox"/>	lps004	sulung kali	19	24.04
<input type="checkbox"/>	lps005	dupak	18	21.13
<input type="checkbox"/>	lps006	simolawang	32	47.00
<input type="checkbox"/>	lps007	pasar kapasan	22	17.37
<input type="checkbox"/>	lps008	tambak rejo	24	125.00
<input type="checkbox"/>	lps009	pecindilan	26	21.45
<input type="checkbox"/>	lps010	simpang dukuh	22	22.45

Percobaan dilakukan dengan menggunakan beberapa parameter seperti yang ditunjukkan dalam Tabel 2:

**Tabel 2:** Parameter Uji Coba

Jumlah LPS	Type Armada	Populasi	Prob. CrossOver	Prob. Mutasi
10	14 m3	100	0.75	0.15

Ketika dilakukan ujicoba selama sepuluh kali, hasil yang didapatkan bisa dilihat dalam Tabel 3:

**Tabel 3:** Hasil Uji Coba

Percobaan Ke-	Jarak yang Dihasilkan	Running Time (detik)	Solusi Pada Generasi Ke-
1	150	56	7
2	145.88	47	5
3	146.4	100	10
4	151.7	125	12
5	145.88	56	7
6	151.57	56	7
7	151.48	54	5
8	148	90	9
9	150.8	140	12
10	153.1	88	9

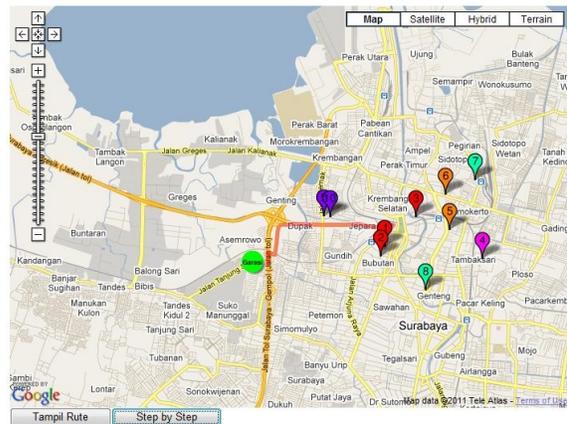
Dari percobaan diatas, bisa dilihat bahwa dari 10 kali percobaan dengan menggunakan data dan parameter sama, algoritma ini bisa memberikan hasil yang berbeda-beda. Hal ini dipengaruhi oleh pembangkitan populasi awal dari Algoritma ini.

Dari percobaan diatas pula, diketahui bahwa hasil terbaik terjadi pada percobaan kelima, dimana didapatkan jarak sebesar 145.88 km, dan mendapatkan solusi pada generasi ke 7. Sedangkan untuk pembagian armada rutennya, bisa dilihat dalam Tabel 4.

**Tabel 4:** Rute Terbaik dari 10 LPS

Kendaraan ke-	Rute yang Diambil
1	Garasi -> lps003 -> lps002 -> lps004 -> LPA
2	Garasi -> lps008 -> LPA
3	Garasi -> lps009 -> lps007 -> LPA
4	Garasi -> lps006 -> lps010 -> LPA
5	Garasi -> lps001 -> lps005 -> LPA

Setelah itu, kita tampilkan hasil terbaik tersebut kedalam Google Maps. Karena memang menampilkan rutennya secara bergantian, maka disini saya hanya meng-capture hanya satu rute saja, dari satu titik ke titik lainnya.



**Gambar 13:** Simulasi Rute dengan Google Maps

Dari hasil ujicoba serta simulasi rute seperti pada Gambar 13 diatas dapat dilihat, bahwa dengan menggunakan sistem optimasi ini, satu armada kendaraan bisa mengambil lebih dari satu LPS, asalkan masih muat dan tidak melebihi kapasitas dari armada tersebut. Dengan demikian, bisa menekan biaya bahan bakar, yang semula harus mengeluarkan biaya untuk 10 truk, menjadi hanya 5 truk saja.

### 8. KESIMPULAN

Dari hasil uji coba sistem ini dapat ditarik beberapa kesimpulan:

1. Algoritma Genetika dapat digunakan untuk menyelesaikan permasalahan dalam pencarian rute terbaik untuk pengangkutan sampah. Sebagai pengembangan dari TSP (Travelling Salesman Problem), Algoritma

Genetika bisa melakukan optimasi rute pengangkutan sampah.

2. Berdasarkan hasil percobaan Algoritma Genetika cocok untuk mendapatkan solusi optimal, namun masih sedikit terkendala dengan beberapa kelemahan. Algoritma ini membutuhkan tiga aspek penting untuk implementasinya yaitu *fitness function*, representasi genetika dan operasi genetika (*crossover* dan *mutation*).
3. Jumlah LPS yang diinputkan oleh user sangat berpengaruh terhadap waktu komputasi dan pencarian solusi terbaik yang bisa dilakukan oleh Algoritma Genetika.
4. Perubahan probabilitas CrossOver dan Mutasi juga mempengaruhi waktu komputasi.
5. Karena ruang solusi yang besar, maka setiap kali melakukan running program, hasil yang didapatkan bisa berbeda-beda.

## 9. REFERENSI

- [1] Wikipedia Indonesia.  
[http://www.id.wikipedia.org/Algoritma\\_genetik](http://www.id.wikipedia.org/Algoritma_genetik).  
21 Januari 2010
- [2] Mangkoedihardjo, Sarwoko, *Peningkatan Kualitas Lingkungan Perkotaan: Pengelolaan Sampah dalam Perspektif Berkelanjutan*, Diskusi Panel Bappenas 12 November 2003, 1 September 2009.
- [3] Kusumadewi, S. & Purnomo, H. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu. 2005.
- [4] Shodiq, Amri. *Tutorial Dasar Pemrograman Google Maps API*. 21 Januari 2011
- [5] Google Maps API Tutorial Eonym.  
<http://eonym.org.uk/gmap/index.htm>. 14 Juli 2011